

# AI-samples

---

本ディレクトリにはamnimo AI Edge Gateway(以下AI GW)上で動作するDNNのサンプルコードが含まれています。

AI GWでは、以下の3つのDNNフレームワークが利用可能です。

- [ONNX Runtime v1.2](#)
- [Tensorflow v1.15.2](#)
- [dv-sdk](#)

AI GWのハードウェアを使用するために、上記の各フレームワークの実装の一部はオリジナルから修正されていますが、使用できるAPIには変更がありません。

## サンプルコード

---

### ort\_tiny\_yolo\_v3.py

ONNX Runtime上でtiny YOLO V3を動作させるサンプルです。

#### 実行前の準備

1. モデルファイルのダウンロード:

```
~/ai-samples-master$ cd model
~/ai-samples-master/model$ wget
https://github.com/onnx/models/raw/master/vision/object_detection_segmentation/tiny-yolov3/model/tiny-yolov3-11.onnx
```

もしAI GWを直接インターネットに接続できない場合は、SCPなどで転送することも可能です。

2. 入力画像の転送

入力画像(フォーマットはJPEGで拡張子は".jpg"である必要があります)を[object\\_detection\\_images/](#)のサブディレクトリ以下に配置してください。例えば[object\\_detection\\_images/images\\_index](#)に記載されている画像をwgetなどで取得することもできます。

#### 実行方法

```
~/ai-samples$ sudo python3 ./ort_tiny_yolo_v3.py
```

- 入力
  - [object\\_detection\\_images/](#)内の画像が使用されます。
- 出力

- 入力画像にバウンディングボックスが付加された画像が`outputs_ort_yolo/`内に出力されます。

## tf\_keras\_mobilenetv2.py

Tensorflow Keras上でMobileNet V2を実行するサンプルです。

### 実行前の準備

入力画像(フォーマットはJPEGで拡張子は".jpg"である必要があります)を`classification_images/`のサブディレクトリ以下に配置してください。例えば`classification_images/images_index`に記載されている画像をwgetなどで取得することもできます。

### 実行方法

```
~/ai-samples$ sudo python3 ./tf_keras_mobilenetv2.py
```

- 入力
  - `classification_images/`内の画像が使用されます。
- 出力
  - 標準出力にクラス名が出力されます。

## cpp\_yolov3\_tiny

dv-sdkを使用してtiny YOLO V3を実行するサンプルです。本サンプルではC++言語を使用しています。

### 実行前の準備

このサンプルでは入力ファイル名が`cpp_yolov3_tiny/main.cpp`にハードコードされているため、`cpp_yolov3_tiny/images/images_index`に記載されているファイルをwgetなどで`cpp_yolov3_tiny/images`にダウンロードしてください。異なるファイルを使用する場合は`cpp_yolov3_tiny/main.cpp`の上部にあるファイル名を変更してください。

### Network Converterを用いたファイル生成

このサンプルのビルドには、設定ファイル(.ini)、Keras標準ファイル(.h5)を入力とし、dv-sdkの一部であるNetwork Converterを用いて生成される"cpp\_yolov3\_tiny\*"という名称のファイル群が必要です。

```
~/ai-samples/cpp_yolov3_tiny$ wget
https://github.com/DigitalMediaProfessionals/application/raw/master/model/yolov3-
tiny.h5
~/ai-samples/cpp_yolov3_tiny$ ls yolov3*
yolov3.ini  yolov3-tiny.h5
~/ai-samples/cpp_yolov3_tiny$ python3 /opt/amnimo-dv720/cnn_converter/convertor.py
yolov3.ini
~/ai-samples/cpp_yolov3_tiny$ ls cpp_yolov3_tiny*
cpp_yolov3_tiny_gen.cpp  cpp_yolov3_tiny_gen.h  cpp_yolov3_tiny_weights.bin
```

Network Converterの入力にはKerasもしくはCaffeのモデルファイルが使用可能です。Network Converter及びdv-sdkの詳細は、[manual](#)や[application](#)を参照してください。

## ビルド及び実行方法:

```
~/ai-samples$ cd cpp_yolov3_tiny
~/ai-samples/cpp_yolov3_tiny$ sudo apt update && sudo apt install build-essential
libopencv-highgui-dev
~/ai-samples/cpp_yolov3_tiny$ ls
cpp_yolov3_tiny_gen.cpp  cpp_yolov3_tiny_gen.h  cpp_yolov3_tiny_weights.bin
images  main.cpp  Makefile  output  YOLOv3_param.h  YOLOv3_post.cpp  YOLOv3_post.h
~/ai-samples/cpp_yolov3_tiny$ make
~/ai-samples/cpp_yolov3_tiny$ sudo ./cpp_yolov3_tiny
```

- 入力
  - `cpp_yolov3_tiny/images/` 内の画像が使用されます。(ファイル名は `cpp_yolov3_tiny/main.cpp` 内にハードコードされています)
- 出力
  - 入力画像にバウンディングボックスが付加された画像が `cpp_yolov3_tiny/output/` 内に出力されます。